Listing of Claims:

Claim 1 (currently amended): A method for enhancing pointer analysis, the method comprising:

processing an assignment between two variables in a program, wherein processing an assignment includes forming a relationship between two locations that are related to the two variables, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein each location includes a label and a content, and wherein a content of a first one of the two locations is selectively unified with a content of ~~the other~~ a second one of the two locations; and

propagating a label of the first one of the two locations to a label of the ~~other~~second one of the two locations such that the label of the first one of the two locations is a subset of the ~~other~~ second one of the two locations.

Claim 2 (original): The method of claim 1, wherein the act of propagating is delayed for a predetermined period of time so as to allow the act of processing an assignment to be executed for each assignment in the program.

Claim 3 (original): The method of claim 1, further comprising forming a points-to graph by iterating the act of processing an assignment for each assignment in the program.

Claim 4 (previously amended): The method of claim 3, wherein forming the points-to graph includes forming a plurality of nodes, and forming a flow line between two nodes so as to represent the relationship between the two locations.

Claim 5 (currently amended): A computer readable medium having instructions stored thereon for causing a computer to perform a method for enhancing pointer analysis, the method comprising:

processing an assignment between two variables in a program, wherein processing an assignment includes forming a relationship between two locations that are related to the two variables, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein each location includes a label and a content, and wherein a content of a first one of the two locations is selectively unified with a content of ~~the other~~ a second one of the two locations; and

propagating a label of the first one of the two locations to a label of the ~~other~~second one of the two locations such that the label of the first one of the two locations is a subset of the ~~other~~ second one of the two locations.

Claim 6 (currently amended): A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

defining a relationship between two locations upon an assignment in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein a label of a first one of the two locations is associated with a label of ~~the other~~ a second one of the two locations, and wherein contents of the two locations are selectively unified.

Claim 7 (currently amended): The method of claim 6, further comprising propagating the label of the first one of the two locations to the ~~other~~ second one of the two locations so as to make the label of the first one of the two locations a subset of the label of the ~~other~~ second one of the two locations.

Claim 8 (currently amended): The method of claim 6, wherein forming the location includes forming a location that points to another location, and wherein the ~~another~~ other location defines a pointed-to location of the location.

Claim 9 (previously amended): The method of claim 8, further comprising defining at least one level of indirection, wherein the at least one level of indirection is defined by at least one location, wherein a pointed-to location of the at least one location defines another level of indirection.

Claim 10 (previously amended): The method of claim 9, wherein defining the relationship includes defining a relationship between the two locations that are in the same level of indirection.

Claim 11 (previously amended): The method of claim 9, wherein defining the relationship includes defining a relationship between the two locations that are in different levels of indirection.

Claim 12 (currently amended): A method of analyzing pointers in a program, the method comprising:

forming ~~a~~ at least one location for at least one variable in the program, wherein the at least one location includes a label and a content; and

forming a relationship between two locations upon an assignment of a first variable and a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein the relationship defines that a label of ~~one~~ a first of the two locations is a subset of a label of ~~the other~~ a second of the two locations, and wherein contents of the two locations are selectively unified.

Claim 13 (currently amended): The method of claim 12, wherein forming the location includes forming the location such that the location points to another location, and wherein the ~~another~~ other location defines a pointed-to location of the location.

Claim 14 (previously amended): The method of claim 13, wherein forming the relationship between two locations includes forming the relationship between two locations upon an assignment of a first variable and a second variable, wherein the second variable is assigned to the first variable.

Claim 15 (currently amended): The method of claim 14, wherein forming the at least one location includes forming a ~~first~~ third location for the first variable and forming a ~~second~~ fourth location for the second variable, wherein the ~~first~~ third location points to the ~~other of the two locations~~first location, and wherein the second location points to the ~~one of the two locations~~ second location.

Claim 16 (currently amended): The method of claim 15, further comprising determining that the program is ~~correctly~~ well typed given that the second variable is assigned to the first variable if and only if a label of a pointed-to location of the second location is a subset of a label of a pointed-to location of the first location, and wherein a content of the pointed-to location of the first location is selectively unified with a content of the pointed-to location of the second location.

Claim 17 (currently amended): A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of an address of a first variable and an address of a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein the relationship defines that a label of first ~~one~~ of the two locations is a subset of a label of a second ~~the other~~ of the two locations, and wherein contents of the two locations are selectively unified.

Claim 18 (currently amended): The method of claim 17, wherein forming the location includes forming the location such that the location points to another location, and wherein the ~~another~~ other location defines a pointed-to location of the location.

Claim 19 (previously amended): The method of claim 18, wherein forming the relationship between two locations includes forming the relationship between two locations upon an assignment of a first variable and a second variable, wherein the second variable is assigned to the first variable.

Claim 20 (currently amended): The method of claim 19, wherein forming the location includes forming a ~~first~~ third location for the first variable and forming a ~~second~~ fourth location for the second variable, wherein the ~~first~~ third location points to the ~~other of the two locations~~first location, and wherein the second location points to the ~~one of the two locations~~ second location.

Claim 21 (currently amended): The method of claim 20, further comprising determining that the program is ~~correctly~~ well typed given that the address of the second variable is assigned to the first variable if and only if a label of the ~~second~~ fourth location is a subset of a label of a pointed-to location of the ~~first~~ third location, and wherein a content of the pointed-to location of the ~~first~~ third location is selectively unified with a content of the ~~second~~ fourth location.

Claim 22 (currently amended): A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of a first variable and a dereference of a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein the relationship defines that a label of ~~one~~ a first of the two locations is a subset of a label of ~~the other~~ a second of the two locations, and wherein contents of the two locations are selectively unified.

Claim 23 (currently amended): The method of claim 22, wherein forming the location includes forming the location such that the location points to another location, and wherein the ~~another~~ other location defines a pointed-to location of the location.

Claim 24 (previously amended): The method of claim 23, wherein forming the relationship between two locations includes forming the relationship between two locations upon an assignment of a first variable and a second variable, wherein the second variable is assigned to the first variable.

Claim 25 (currently amended): The method of claim 24, wherein forming the location includes forming a ~~first~~ third location for the first variable and forming a ~~second~~ fourth location for the second variable, wherein the ~~first~~ third location points to the ~~other~~ first of the two locations, wherein the ~~second~~ fourth location points to a first pointed-to location, and wherein the first pointed-to location points to the ~~one~~ second of the two locations to define a second pointed-to location.

Claim 26 (currently amended): The method of claim 25, further comprising determining that the program is ~~correctly~~ well typed given that the dereference of the second variable is assigned to the first variable if and only if a label of the second pointed-to location is a subset of a label of a pointed-to location of the first location, and wherein a content of the pointed-to location of the ~~first~~ third location is selectively unified with a content of the second pointed-to location.

Claim 27 (currently amended): A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of a dereference of a first variable and a second variable in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein the relationship defines that a label of ~~one~~ a first of the two locations is a subset of a label of ~~the other~~ a second of the two locations, and wherein contents of the two locations are selectively unified.

Claim 28 (currently amended): The method of claim 27, wherein forming the location includes forming the location such that the location points to another location, and wherein the ~~another~~ other location defines a pointed-to location of the location.

Claim 29 (previously amended): The method of claim 28, wherein forming the relationship between two locations includes forming the relationship between two locations upon an assignment of a first variable and a second variable, wherein the second variable is assigned to the first variable.

Claim 30 (currently amended): The method of claim 29, wherein forming the location includes forming a ~~first~~ third location for the first variable and forming a ~~second~~ fourth location for the second variable, wherein the ~~first~~ third location points to a pointed-to location that points to the ~~other~~ second of the two locations to define a first pointed-to location, wherein the ~~second~~ fourth location points to the ~~one~~ first of the two locations.

Claim 31 (currently amended): The method of claim 30, further comprising determining that the program is ~~correctly~~ well typed given that the second variable is assigned to the dereference of the first variable if and only if a label of a pointed-to location of the ~~second~~ fourth location is a subset of a label of the first pointed-to location, and wherein a content of the first pointed-to location is selectively unified with a content of the pointed-to location of the ~~second~~ fourth location.

Claim 32 (currently amended): A computer readable medium having instructions stored thereon for causing a computer to perform a method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

defining a relationship between two locations upon an assignment in the program, wherein the two locations are selected to be one level of indirection away from a level associated with the assignment, wherein a label of ~~one~~ a first of the two locations is defined as a subset of a label of ~~the other~~ a second of the two locations, and wherein contents of the two locations are selectively unified.

Claim 33 (previously amended): The method of claim 32, wherein defining the relationship includes defining the relationship between the two locations upon an assignment of a first variable and a second variable in the program, wherein the first variable and the second variable are pointers.

Claim 34 (previously amended): The method of claim 32, wherein defining the relationship includes defining the relationship between the two locations upon an assignment of a first variable and an address of a second variable in the program.

Claim 35 (previously amended): The method of claim 32, wherein defining the relationship includes defining the relationship between the two locations upon an assignment of a first variable and a dereference of a second variable in the program.

Claim 36 (previously amended): The method of claim 32, wherein defining the relationship includes defining the relationship between the two locations upon an assignment of a dereference of a first variable and a second variable.

Claims 37-51 (canceled)

Claim 52 (previously amended): A method for enhancing pointer analysis, the method comprising:

processing a plurality of assignment statements in a program to derive a plurality of sets of information, wherein the plurality of sets of information is distributed among a plurality of levels of indirection;

establishing a plurality of flow relationships corresponding to each of the plurality of assignment statements, wherein each of the flow relationships is selected to be established one level of indirection away from each of the assignment statements; and selectively unifying at least one of the sets of information in at least one level of indirection so as to allow a desired level of analytical precision within a desired duration of pointer analysis.

Claim 53 (previously amended): The method of claim 52, wherein the act of selectively unifying includes unifying sets of information in all levels of indirection except for a first level of indirection.

Claim 54 (currently amended): The method of claim 52, wherein the desired duration of pointer analysis is ~~about~~ linearly proportional to the size of the program.

Claim 55 (previously amended): A system for enhancing pointer analysis of a program, wherein the program includes at least one source file, the system comprising:

a compiler to compile the at least one source file to produce an intermediate language;

a builder receptive to the intermediate language to build a tree that represents the at least one source file; and

an analyzer to analyze the tree to produce an object file, wherein the object file contains at least one relationship between two variables in an assignment statement in the program, wherein the relationship defines that a set of symbols relating to one of the two variables is a subset of a set of symbols relating to the other of the two variables, and wherein another relationship is selectively formed one level of indirection away from a level associated with the assignment statement between the set of symbols related to one of the two variables and the set of symbols relating to the other of the two variables.

Claim 56 (original): The system of claim 55, further comprising a linker to link a plurality of object files of the program so as to produce a pointer analysis for the program.